A 28nm 97.23 TFLOPS/W Pipelined Two-Stage-Aggregation Compute-in-Memory Macro for Edge-LLM Inference and Fine-tuning

*Yi Yang[1], *Jinwu Chen[1], Yutong Zhang[1], Yucheng Du[1], Defa Wu[1], Yingxuan Zhou[1], Yanqi Zhang[1], Tianhui Jiao[1], Xiaoxue Zhong[1], Xing Wang[1], Zhichao Liu[1], Yuchen Tang[1], An Guo[1,2], and Xin Si[1]
[1]Southeast University, Nanjing, China; [2]HOUMO AI, Shanghai, China
*Equally contributed authors; Corresponding email: xinsi@seu.edu.cn

***Abstract*—This work presents a 28nm eDRAM compute-in-memory (CIM) macro featuring a Two-Stage-Aggregation computation flow to mitigate poor weight reuse and prohibitive runtime dequantization overhead in edge LLM applications, achieving 8.7× energy efficiency and 9.1× area efficiency over traditional dequantize-first methods. The main contributions include: 1) A ping-pong output-stationary node for routing-based local accumulation; 2) A tiered computation mapping with a shared global dot-product; 3) A node-splitting method for single-BF16/dual-INT8 reconfiguration. The fabricated macro achieves measured peak energy efficiency of 97.23 TFLOPS/W and area efficiency of 2.08 TFLOPS/mm^2.**

***Index Terms*—Compute-in-memory, eDRAM, edge LLM, non-uniform quantization, Two-Stage-Aggregation, reconfigurable.**

I. INTRODUCTION

Compute-in-memory (CIM) architectures have shown high energy efficiency for CNNs by minimizing data movement [1]–[8], yet they struggle with edge LLM workloads as shown in Fig. 1. Specifically, conventional weight-stationary (WS) CIMs suffer severe efficiency degradation due to poor weight reuse in low-batch edge LLM inference. Furthermore, while memory-efficient non-uniform quantization (NUQ) formats effectively reduce model size [9], [10], they incur high logic overhead due to runtime dequantization and complex floating-point multiply-accumulate (MAC), which traditional CIM architectures fail to handle efficiently.

To overcome these limitations of conventional CIM architectures, this work proposes an output-stationary (OS) CIM featuring a *Two-Stage-Aggregation* (TSA) MAC flow as shown in the upper portion of Fig. 2, which decouples the computation into two distinct stages:

- **Stage-1** An intra-group accumulation (IGA) stage, where high-precision activations are grouped according to their corresponding W_{index} and aggregated within each group;
- **Stage-2** An inter-group dot product (IGDP) stage, where these accumulated sums are aggregated by their corresponding W_{value} to produce the final output.

Unlike conventional dequantize-first method where dequantization and multiplication overhead scales linearly with the MAC length (N), the proposed TSA method decouples these resource-intensive operations from the primary accumulation chain. The computation complexity is reduced from $O(N)$ to $O(1)$ relative to N, effectively cutting the operation count of typical MAC workload by over 80%.

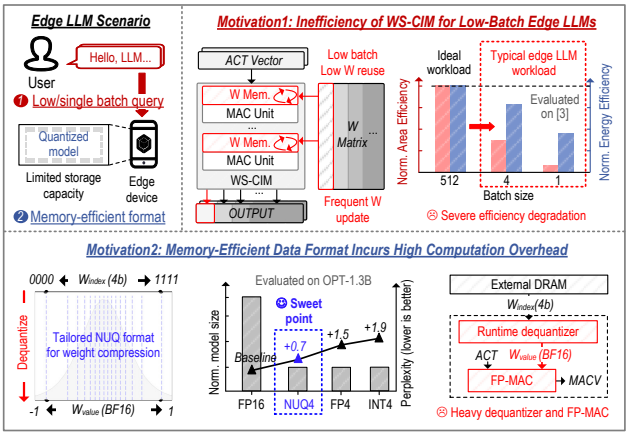


Fig. 1. Inefficiencies of conventional CIM architectures for tasks of edge LLM scenario.

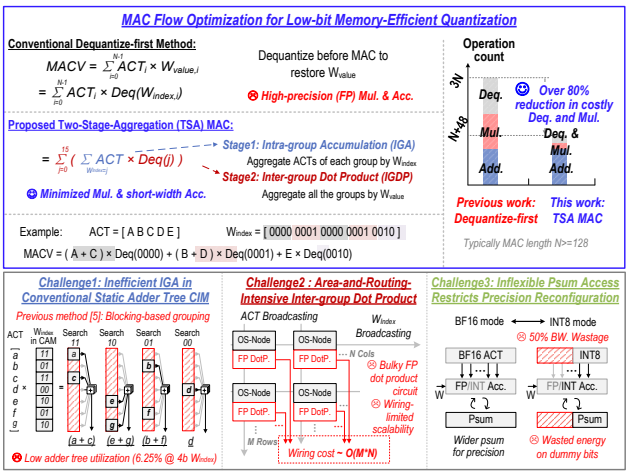


Fig. 2. Two-Stage-Aggregation MAC flow and circuit-level challenges.

However, as illustrated in the lower portion of Fig. 2, efficiently mapping the proposed TSA MAC flow into hardware

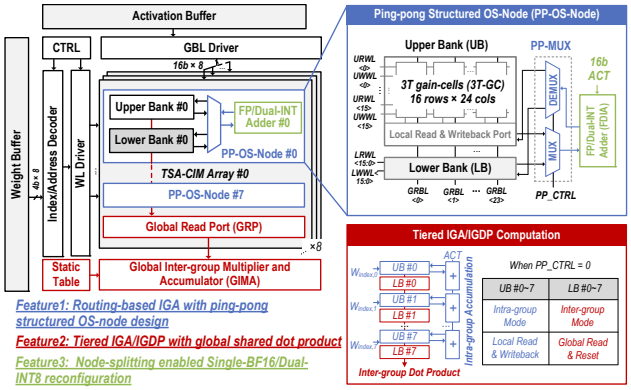


Fig. 3. Overall architecture and three key features of proposed TSA-CIM.

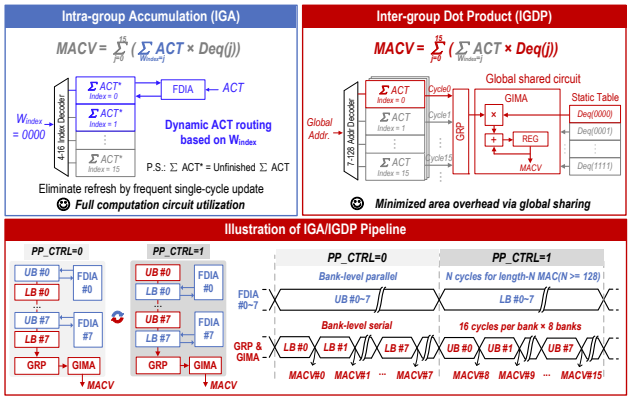


Fig. 4. Illustration of pipelined IGA/IGDP.

presents three major circuit-level challenges. First, the inherent sparsity of IGA results in extremely low hardware utilization within conventional static adder-tree-based CIMs. Specifically, computation utilization drops exponentially as the bit-width of W_{index} increases [5]. Second, the bulky floating-point logic required for IGDP introduces severe area overhead and an unscalable $M \times N$ global wiring cost. Third, fixed-width local memory causes inflexible data access, restricting efficient precision reconfiguration and limiting the macro's adaptability to diverse workloads, such as inference and fine-tuning.

To overcome these bottlenecks, this work proposes a 48kb TSA-CIM macro featuring three key features: 1) A ping-pong structured output-stationary node (PP-OS-Node) design that enables routing-based IGA, maximizing hardware utilization; 2) A tiered IGA/IGDP mapping strategy with bank-level shared IGDP module; and 3) A node-splitting technique for efficient single-BF16/dual-INT8 reconfiguration, ensuring high adaptability across diverse workloads. Fabricated in 28nm, the proposed macro achieves a peak energy efficiency of 97.23 TFLOPS/W and an area efficiency of 2.08 TFLOPS/mm^2 in high-efficiency mode.

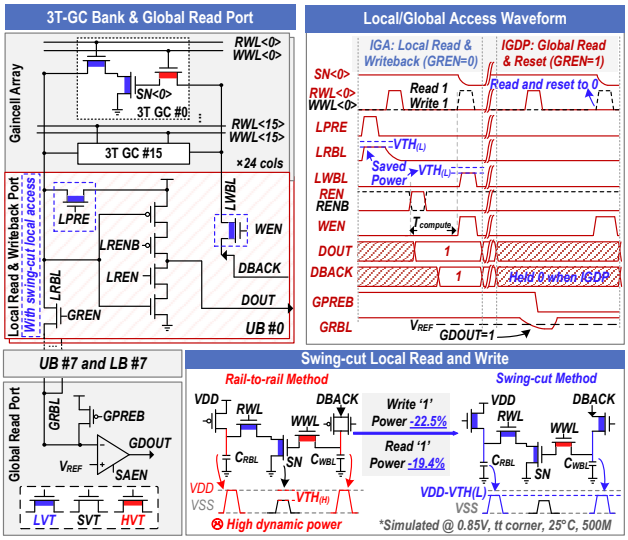


Fig. 5. Details of 3T gain-cell bank and swing-cut local access.

II. PROPOSED TSA-CIM MACRO DESIGN

Fig. 3 illustrates the overall architecture of the proposed TSA-CIM macro. It comprises eight arrays, with each array integrating eight ping-pong-structured OS-Nodes (PP-OS-Nodes), a global read port (GRP), and a global inter-group multiplier and accumulator (GIMA). This 8×8 arrangement orthogonally aligns eight W_{index} inputs along the word-lines (WLs) and eight activation inputs along the bit-lines (BLs), enabling the parallel computation of 64 output channels. As detailed in the node view, each PP-OS-Node features two 16×24 3T gain-cell (3T-GC) banks—an upper bank (UB) and a lower bank (LB)—coupled with a ping-pong multiplexer (PP-MUX) and a reconfigurable FP/dual-INT adder (FDIA).

A dedicated control signal PP_CTRL dynamically toggles the banks between IGA and IGDP modes. For example, when PP_CTRL = 0, all the UBs operate in the IGA mode, performing local read-out and write-back. Simultaneously, all LBs are configured for IGDP mode, performing global read-out and reset.

A. Intra-group Accumulation and Inter-group Dot Product

Fig. 4 details the execution of the two-stage TSA computation flow. The upper-left inset depicts the local routing-based IGA, where a 4-to-16 index decoder dynamically routes incoming activations to the specific accumulator row corresponding to their W_{index}. The upper-right illustrates the global shared IGDP stage. The pre-accumulated group sums are sequentially read out via the GRP, scaled by the dequantized W_{value} retrieved from a pre-defined static table, and aggregated within the shared GIMA to produce the MAC value (MACV).

The lower portion demonstrates the pipelined IGA/IGDP dataflow. While the UBs perform parallel IGA, the LBs execute sequential global readout. Upon completion, the banks swap roles under the ping-pong control PP_CTRL: the UBs transition to the readout phase, whereas the LBs are reset for

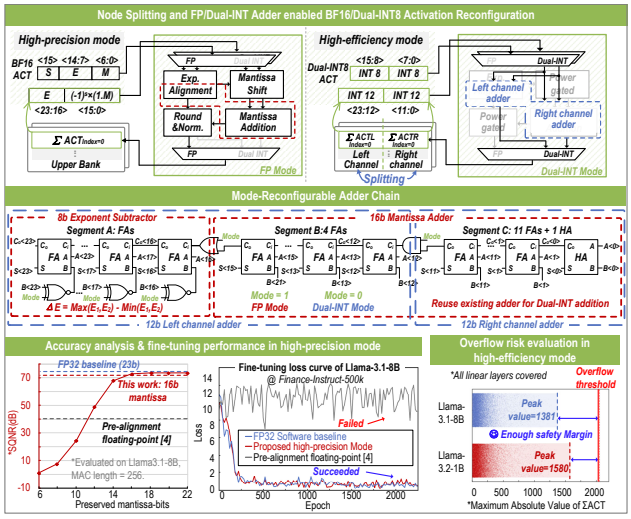


Fig. 6. Single-BF16/Dual-INT8 reconfiguration and real-task evaluation

the next IGA cycle. Since the accumulation length (N) in typical LLM applications generally exceeds 128, this pipelined approach ensures that the key compute units, the local FDIAs, remain fully saturated. The shared GIMA processes only one row of accumulated results from the 3T GC banks each cycle, thereby achieving minimal area overhead. Furthermore, this sequential GRBL readout mechanism effectively balances the internal bandwidth pressure within the macro. This interleaved pipelining strategy fully leverages the area-intensive local FDIAs while effectively amortizing the shared GIMA workload and optimizing memory throughput.

B. 3T Gain-cell Bank Details and Swing-cut Local Access

Fig. 5 illustrates the circuit implementation of the 3T gain-cell bank along with its detailed local/global access mechanisms. During the IGA phase, the local port executes a high-speed local read-and-writeback sequence. In the subsequent IGDP phase, the control logic transitions to the global port to perform a global read-and-reset operation, where the stored data is fetched and the cell is concurrently cleared to zero.

To mitigate the dynamic power overhead arising from frequent local accesses, a swing-cut local access scheme is proposed to suppress the voltage swings on the local read bit-line (LRBL) and local write bit-line (LWBL). In a conventional rail-to-rail architecture, the LRBL is pre-charged to V_{DD} and fully discharged to ground during a Read-'1' operation. Furthermore, while a high-threshold voltage (HVT) access transistor inherently limits the storage node (SN) voltage to $V_{DD} - V_{TH(H)}$ during a Write-'1' write-back, the LWBL is still unnecessarily driven to the full V_{DD} level, leading to power wastage. The proposed swing-cut scheme addresses this inefficiency by employing low-threshold voltage (LVT) NMOS transistors for both the read pre-charge and the write-back access gates. This configuration leverages the $V_{TH(L)}$ drop of the LVT devices to naturally clamp the voltage swings on the LRBL and LWBL, thereby maintaining robust operation while significantly enhancing energy efficiency. Evaluation results demonstrate that the proposed swing-cut method reduces write '1' and read '1' power by 22.5% and 19.4%, respectively.

C. Node-splitting Enabled Single-BF16/Dual-INT8 Reconfiguration

Fig. 6 illustrates the proposed node-splitting scheme, which enables reconfigurable support for single-BF16 or dual-INT8 operations within a fixed partial sum (Psum) width. In high-precision (HP) mode, the node functions as a unified channel, accumulating a single BF16 activation into a 24-bit Psum. Conversely, in high-efficiency (HE) mode, the node splits into dual channels by partitioning the Psum into two 12-bit segments, allowing for the parallel accumulation of two INT8 activations. A key feature is the mode-reconfigurable adder chain, which leverages carry-chain segmentation to repurpose existing floating-point (FP) logic—specifically the 8-bit exponent subtractor and 16-bit mantissa adder—into dual 12-bit integer adders, thereby incurring no additional adder hardware overhead. Furthermore, unused FP components are power-gated to eliminate leakage, providing energy-efficient adaptability for both high-precision fine-tuning and high-throughput inference tasks.

The lower portion of Fig. 6 presents the real-task evaluation results regarding overflow and accuracy. Overflow analysis on various edge-scale models confirms a sufficient safety margin for the 12-bit Psum within the HE mode, ensuring robust accumulation. Furthermore, accuracy benchmarks demonstrate that the HP mode achieves parity with the FP32 baseline. This capability successfully enables LLM fine-tuning tasks where conventional pre-alignment methods fail to converge [4].

III. MEASUREMENT RESULTS

Fig. 7 quantifies the architectural benefits of the proposed schemes and provides the Shmoo plots for both HP and HE modes. Compared to the dequantize-first method and the blocking-based method [5], our proposed routing-based IGA with a shared IGDP circuit achieves 8.7× and 7.3× higher energy efficiency, as well as 9.1× and 8.3× higher area efficiency, respectively. Compared to traditional single-FP/single-INT reconfiguration, the proposed single-FP/dual-INT scheme achieves 2.0× higher throughput, 1.8× higher area efficiency, and 1.6× higher energy efficiency in HE mode.

Fig. 8 presents the die micrograph, measured voltage-frequency curves, and the chip summary table. The 48kb TSA-CIM macro, fabricated in 28nm CMOS, occupies a compact area of 0.112 mm^2 with a memory density of 428.57 kb/mm^2. In high-efficiency mode, the macro achieves a peak energy efficiency of 97.23 TFLOPS/W and a peak area efficiency of 2.08 TFLOPS/mm^2. Task-level evaluations on both inference and fine-tuning tasks demonstrate minimal degradation compared to the software baseline, confirming the precision robustness of the proposed architecture.

Fig. 9 compares this work with state-of-the-art (SOTA) CIM macros and accelerators. The 48kb output-stationary macro integrates dequantization and MAC functions, providing native

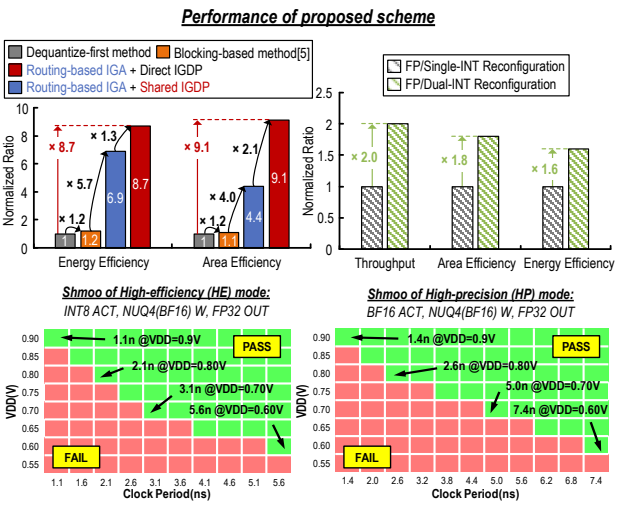
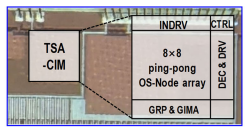
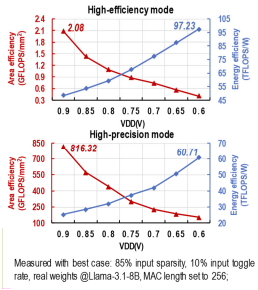


Fig. 7. Performance evaluation and measured shmoo plots.





Measured with best case: 85% input sparsity, 10% input toggle rate, real weights @Llama-3.1-8B, MAC length set to 256;

CHIP SUMMARY	
Technology	28nm CMOS
CIM Type	Digital
CIM Dataflow	Output Stationary
Supply Voltage	0.6V-0.9V
Cell	3T gain-cell (Logic rule)
Macro Size	48kb
Macro Area	0.34mm × 0.33m = 0.112 mm²
Input precision	BF16,INT8
Weight precision	NUQ4,INT8
Output precision	FP32
Area Efficiency (TFLOPS/mm²)	BF16 × NUQ4: 0.15*1~0.82*2 INT8 × NUQ4: 0.41*1~2.08*2
Energy Efficiency*3 (TFLOPS/W)	BF16 × NUQ4: 19.36*2~60.71*1 INT8 × NUQ4: 31.45*2~97.23*1
Memory Density (kb/mm²)	428.57
Inference Perplexity*3 (Lower is better)	Llama-3.2-1B: 17.862 (+0.063) Llama-3.1-8B: 9.227 (+0.017)
Fine-tuning Quality*4 (Perplexity on target task)	1.411 (+0.002)

*1 Measured with best case at 0.6V: 85% input sparsity, 10% input toggle rate, real weights @Llama-3.1-8B, MAC length set to 256;
*2 Measured with average case at 0.9V: 65% input sparsity, 30% input toggle rate, real weights @Llama-3.1-8B, MAC length set to 256;
*3 Evaluated on Wikitext2-raw-v1 in high-efficiency mode, using software inference perplexity as baseline;
*4 Evaluated @Llama-3.1-8B fine-tuned on Finance-Instruct-500k in high-precision mode, using software fine-tuned model as baseline.

Fig. 8. Die photo, voltage-efficiency curve and chip summary.

support for NUQ weights and accommodating the diverse precision demands of edge inference and fine-tuning. Compared to prior CIM macros [1]–[4], [8], this work improves the figure-of-merit (FoM) by 2.30 – 5.63× and 1.13 – 2.77× in HE and HP modes, respectively.

IV. CONCLUSION

This paper presents a 28nm output-stationary CIM macro featuring a TSA computation flow tailored for efficient edge LLM inference and fine-tuning. A PP-OS-Node architecture with routing-based IGA is proposed, maximizing circuit utilization through pipelined execution. A tiered intra/inter-group computation mapping is employed, achieving an area-efficient inter-group IGDP without compromising throughput. A dynamic node-splitting scheme provides flexible reconfiguration between single-BF16/dual-INT8 to accommodate diverse applications. The 48kb test chip, fabricated in 28nm CMOS, achieves a peak energy efficiency of 97.23 TFLOPS/W and a peak area efficiency of 2.08 TFLOPS/mm^2.

	ISSCC'25[1]	ESSERC'25[2]	ISSCC'24[3]	ISSCC'23[4]	CICC'25[8]	This work	
Technology	28nm	28nm	28nm	28nm	28nm	28nm	
Cell	SRAM	4T2C	SRAM	SRAM	SRAM	3T Gain-cell	
Retention	\	167u	\	\	\	61.2u	
Area (mm^2)	0.14	12.96	1.94	0.15	0.37	0.11	
Macro Size (kb)	64	2440	192	64	32	48	
Voltage (V)	0.55-0.9	1	0.7-0.95	0.6-0.9	0.45-1	0.6-0.9	
Compute Dataflow	WS	WS	WS	WS	OS	OS	
Task	Inference	Inference	Inference	Inference	Inference	Inference, Fine-tuning	
CIM Function	MAC	MAC	MAC	MAC	MAC	Dequantize + MAC	
Weight Precision	BF16	2~8b	BF16	BF16	INT 8	NUQ4(BF16), INT8	
Activation Precision		8b				BF16, INT8	
Output Precision	FP32	\	8b	BF16	8-16b	FP32	
Energy Efficiency*1 (TFLOPS/W)	17.83-62.84	47.5	16.55-32.78	14.04-31.6	6.7-40.2	High-efficiency: 31.45-97.23*2	High-precision: 19.36-60.71*2
Area Efficiency*1 (TFLOPS/mm^2)	0.70	\	2.21	2.56	1.60	0.41-2.08*3	0.15-0.82*3
FoM*4 (×10^4)	36.02	\	14.7	33.13	3.29	82.83	40.78

*1. A length-N MAC counts as 2N-1 operations; *2. Peak energy efficiency measured at 0.6V, 130M; *3. Peak area efficiency measured at 0.9V; *4. FoM = Activation Precision × Weight Precision × Output Precision × Energy Efficiency × Area Efficiency ;

Fig. 9. Comparison with state-of-the-art CIM works.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 62522403); by the National Key R&D Program of China (Grant No. 2022ZD0118902); by the National Natural Science Foundation of China (Grant Nos. 92264203, 92464202, 92464302); by the Key R&D Program of Jiangsu Province (Grant No. BE2023020-1); in part by the Fundamental Research Funds for the Central Universities; and in part by Xiaomi Xuanjie.

REFERENCES

[1] X. Wang *et al.*, "A 28nm 17.83-to-62.84TFLOPS/W Broadcast-Alignment Floating-Point CIM Macro with Non-Two's-Complement MAC for CNNs and Transformers," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 254–256, 2025.

[2] H. Jeong, *et al.*, "A 30.7 TOPS/W Sparsity-Aware Analog-Digital Hybrid eDRAM CIM by Effective Row Activation with Simultaneous Multi-Row-Multi-Task Control," *IEEE Eur. Solid-State Electron. Res. Conf. (ESSERC)*, pp. 117–120, 2025.

[3] Y. Yuan *et al.*, "A 28nm 72.12TFLOPS/W Hybrid-Domain Outer-Product Based Floating-Point SRAM Computing-in-Memory Macro with Logarithm Bit-Width Residual ADC," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 576–578, 2024.

[4] A. Guo *et al.*, "A 28nm 64-kb 31.6-TFLOPS/W Digital-Domain Floating-Point-Computing-Unit and Double-Bit 6T-SRAM Computing-in-Memory Macro for Floating-Point CNNs," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 128–130, 2023.

[5] Z. Wu *et al.*, "CELLA: A 28nm Compute-Memory Co-Optimized Real-Time Digital CIM-Based Edge LLM Accelerator with 1.78ms-Response in Prefill and 31.32 Token/s in Decoding," *IEEE Symp. VLSI Technol. Circuits (VLSI Symp.)*, pp. 1–3, 2025.

[6] W.-S. Khwa *et al.*, "A 16nm 96Kb Integer/Floating-Point Dual-Mode-Gain-Cell-Computing-in-Memory Macro Achieving 73.3-163.3TOPS/W and 33.2-91.2TFLOPS/W for AI-Edge Devices," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 568–570, 2024.

[7] A. Guo *et al.*, "A 22nm 64kb Lightning-Like Hybrid Computing-in-Memory Macro with a Compressed Adder Tree and Analog-Storage Quantizers for Transformer and CNNs," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 570–572, 2024.

[8] X. Hu *et al.*, "A 28nm 20.9-137.2 TOPS/W Output-Stationary SRAM Compute-in-Memory Macro Featuring Dynamic Look-ahead Zero Weight Skipping and Runtime Partial Sum Quantization," *IEEE Custom Integr. Circuits Conf. (CICC)*, pp. 1–3, 2025.

[9] T. Dettmers *et al.*, "QLORA: Efficient Finetuning of Quantized LLMs," *Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 10088–10115, 2023.

[10] J. Dotzel *et al.*, "Learning from Students: Applying t-Distributions to Explore Accurate and Efficient Formats for LLMs," *Int. Conf. Mach. Learn. (ICML)*, pp. 11573–11591, 2024.